REMARKS/ARGUMENTS

These remarks are made in response to the Office Action of December 08, 2003 (Office Action). As this response is timely filed within the 3-month shortened statutory period, no fee is believed due.

In paragraphs 1 and 2 of the Office Action, FIG. 1 was objected to under 37 C.F.R. § 1.84 (p)(5) because the reference character "30" did not appear in the description. FIG. 4 was similarly objected to because the reference character "410c" did not appear in the description. In response, Applicants have amended the description to include the reference character "30" and "410c." Applicants believe this amendment corrects the deficiencies and that no drawing correction is required. Accordingly, Applicants respectfully request that the paragraph 1 and 2 objections be withdrawn.

In paragraph 3, 4, and 5 of the Office Action, the abstract was objected to for purporting merits or speculative applications of the invention, for exceeding 150 words, and for including minor grammatical informalities. Applicants have replaced the abstract to overcome these objections. Accordingly, Applicants respectfully request that the paragraph 3, 4, and 5 objections be withdrawn.

In paragraph 6 of the Office Action, claims 1-33 have been rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent Number 5,261,044 to Dev *et al.* (Dev).

In response to the Office Action, Applicants have amended claim 1 to incorporate the limitations previously specified within dependent claims 2 and 3 within independent claim 1. Additionally, Applicants have amended claim 9 to incorporate the limitations previously specified within dependent claims 10 and 11. Corresponding apparatus claims 16 and 24 have been similarly amended. Claims 2, 3, 10, 11, 17, 18, 25, and 26 have been canceled. Claims 4, 12, 19, and 27 have been amended to maintain proper dependencies in light of the above amendments.

Claim 32 has been amended to clarify that the reporting system includes a plurality of components distributed across a heterogeneous network, as specified in FIGS 1, 3, and the corresponding detailed description. Claim 32 has also been amended to clarify that individual

ones of the nodes can include a plurality of visualizations, as specified in FIG. 2 and the corresponding detailed description. Finally, claim 32 has been modified to clarify that reported metric values displayed within the graphical display map are based upon component metric values gathered by agents, as specified in FIG. 1 and the corresponding detailed description.

Claim 34 has been added to detail that agents can be software objects, as specified on page 10, second paragraph as well as in FIG. 1 and the corresponding detailed description. Claim 35 has been added to clarify that the agents can be distributed across the heterogeneous network, as specified on page 10, second paragraph as well as in FIG. 3 and the corresponding detailed description. Claim 36 has been added to detail that a node can include multiple visual representations, each representing a component metric value, as specified in FIGS. 2 and 5 and the corresponding detailed description. Claim 37 has been added to detail that different agents can provide metrics for a node, as specified in FIG. 4 and the corresponding detailed description. Claim 38 has been added to detail that reported metric values can include metric values and/or codes, as specified in FIGS. 4 and FIG. 5 and the corresponding detailed description. Claim 33 has been canceled. No new matter has been added because of these amendments.

Prior to addressing the rejections on the art, a brief review of the Applicants' invention is in order. The Applicants claimed and disclosed subject matter teaches a system, method, and apparatus for dynamically exposing the nodes of a graphical display or a display map. Each of the nodes can present data metrics for a multitude components, where the components can be distributed across a network. Monitored components can include both hardware and software components that utilize vastly different computing platforms, communication protocols, performance metrics, and the like. A multitude of different monitoring bots, or autonomous software agents, can be used as intermediaries between the nodes of the display map and the monitored components. A node can display information gathered by a multitude of different software agents.

Software agents can convey messages to nodes, each message containing a multitude of reported metrics. Each reported metric can be either a metric value or a code. The metric value can be converted into a visual representation associated with the metric and displayed within the

display map. For example, a metric of 90% utilization can be visually represented by a bar chart, by a gauge, by a text value, by a pie-chart, and/or the like.

The code can result in a designated programmatic action. For example, a code of "-1" can disable an associated visual representation and/or place a not-applicable indicator in an appropriate display location. A code of "-2" can indicate a communication problem has occurred. A code of "-3" can indicate that a different software agent is to supply the associated metric. That is, a code of "-3" can be a placeholder indicating that no metric data for a particular attribute is being provided.

By way of example, a node can have five metric values associated with it. These values can be provided to the node by a first and a second software agent. The first software agent can provide the first and fifth values and the second software agent can provide the second, third, and fourth values. Using the aforementioned coding conventions, a message can be sent by the first software agent containing five elements, [value1, -3, -3, -3, value5], and a different message can be sent by the second software agent containing five elements, [-3, value2, value3, value4, -3]. When the message from the first software agent is received, value1 and value5 (considered determined metric values) can be used to update the node within the display map. At this point the second, third, and forth values are considered to be undetermined metric values. When the message from the second software agent is received, the previously undermined metric values can now be determined from the message and the appropriate updates can be made to the display map.

Turning specifically to the rejections on the art, in paragraph 6 of the Office Action, claims 1-33 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Dev. Dev discloses a layered network management system. The layers include a user interface 10, a virtual network machine 12, a device manager 14, and a network 18 containing a multitude of monitored network components. All devices in the network layer 18 convey information to the device communication manager 14. The device communication manager 14 converts the communication protocol of network devices into a standardized protocol established for the network management system. The virtual network machine 12 is a network management server that can centrally gather device information, store this information in a database 16, and serve

network information to one or more user interfaces 10. Each of these user interfaces can have interface specific settings via the view personality module 20.

Referring to claims 1, 9, 16, and 24, the Examiner has asserted that the virtual network machine 12 of Dev as recited in column 7, lines 25-44 is equivalent to the first agent claimed by the Applicants. The Applicants respectfully disagree. The virtual network machine 12 is defined as a server that centrally manages network data, whereas the Applicants' agents are decentralized autonomous software objects distributed at suitable locations within a heterogeneous environment. Each software agent can be configured to monitor a particular component to which the agent has been assigned. Accordingly, the virtual network machine 12 is a complex, computationally heavy, centralized server, whereas a software agent is a discrete software unit uniquely constructed to monitor selected component attributes in a dedicated and efficient manner.

For example, the capabilities of the virtual network machine 12 are so robust that the virtual network machine 12 can track all monitored aspects of a regional network across vast geographical regions and provide capabilities to monitor a network using different levels of granularity, as shown in FIG. 7A, FIG. 7B, FIG. 7C, FIG. 8A, FIG. 8B, FIG. 9, and FIG. 10. The robust features contained within the virtual network machine are emphasized throughout Dev. For example, at column 4, lines 9-16, Dev states:

> The virtual network machine 12 contains a software representation of the network being managed, including models that represent the devices and other entities associated with the network, and relationships between models. The virtual network machine is associated with a database manager 16 which manages the storage and retrieval of disk based data.

In contrast to the virtual network machine 12, the first agent of claim 1 is a software agent configured to perform a dedicated data gathering function. The software agent (as specified on page 11, line 9) is a small, autonomous software object having a well defined monitoring and data conveyance task. As used herein and as defined by the Cyber Business Center (TM) accessible via **http://www.nottingham.ac.uk/cyber/G051.html**, a software agent is:

A "smart" computer program (or infomachine) that can "serve" its human "master" in cyberspace. Software agents (sometimes also known as bots protect their users from the complexity of computer and network operations, and may engage in database searches and transactions based upon a knowledge of an evolving user profile.

The decentralized nature of a software agent is further emphasized in claim 1, by having a first agent and a second agent independently functioning as intermediaries, as specified in FIG. 3. Node values handled by the first agent are referred to as the "determined metric values" that are initially determined when the first agent reports data to the graphical display interface. Node values handled by the second agent are referred to as the "undetermined metric values" that are not determined until the second agent reports data to the graphical display interface.

Additionally, the device communication manager 14 of Dev is cited in the Office Action as being equivalent to the second agent claimed by the Applicants, which is respectfully refuted. The device communication manager 14 (as defined at column 4, lines 18) handles communication between the virtual network machine 14 and the network devices. That is, the device manager translates component communications into a standardized format defined for the network management system of Dev. The device communication manager 14 is not described as having the ability to probe components, to determine metrics, or to report metrics to a graphical display unit. In contrast, the Applicants' second agent is a software agent having similar capabilities as the first software agent, allowing the second software agent to perform the claimed probing and reporting steps.

Claims 4, 12, 19, and 29 specify that a message containing metric values and a predefined code can be conveyed by the software agent to the display map. Column 9, lines 8-39 of Dev is cited as anticipating these claims. Dev, however, teaches that all network modeling and component attributes are to be performed within a virtual network machine that servers models to client interfaces. The cited portion of Dev teaches that exception handling (error rate interference handling and fault isolation inference handling) can be performed when monitored components experience problems. These exception handling procedures can produce any programmatic results, such as producing a chain of actions or responses to the attribute change, as per column 9, lines 51-53 of Dev. Exception handling is performed within the virtual network

machine functioning as a server. Dev is silent with regard to passing values and control codes to a client for display purposes. Also, as previous noted, Dev is silent as to the utilization of software agents or bots.

In contrast, the Applicants teach that multiple determined metric values and code values can be conveyed within a message from a software agent to a display map. These messages are used to by the display map to update display values and to synchronize data arriving from multiple software agents. The Applicants have selected the claimed data passing method to minimize the number of messages that are needed to convey data to the display map, thereby improving response time and reducing network overhead. Dev fails to provide any such teaching.

Regarding claim 32, the icons detailed at column 2, lines 47-50 of Dev are equated to the nodes within the Applicants' submission. An icon is a simple graphical image, where the nodes described by the Applicants are can be more robust display mechanisms. Selective ones of the nodes can, for example, include many different visual representations, such as a table of values or multiple text boxes, as specified in FIG. 2 and the corresponding detailed description. That is, a node, unlike an icon, can include multiple attributes of information conveyed from multiple software agents.

Additionally, column 3, lines 1-28 and column 4, lines 14-27 of Dev have been cited as teaching a plurality of agents. The cited portions of Dev, however, are silent in regard to software agents. Instead, Dev teaches at column 4, lines 9-13 that a centralized server called the virtual network machine is to be used to drive a network management system.

Not only is Dev silent as to the use of software agents to function as intermediaries between a display map and a plurality of components, but use of software agents would contrast the teaching of Dev. That is, Dev teaches a centralized server through which all traffic between components and user interfaces is routed, while the Applicants teach a decentralized, flexible, modular methodology where data can flow between the components and the display map via a plurality of pathways. These data pathways generally use a software agent as a customized communication intermediary.

Regarding claim 34, Dev fails to teach the use of software agents.

Regarding claim 35, Dev teaches a centralized server for network management, not a system where software agents that function as intermediaries between components and a display map can be distributed across a heterogeneous network.

Regarding claims 36 and 37, all user interface updates occur through a centralized location in Dev. Dev fails to teach that many visual representations or attributes can be presented for each node. Dev also fails to teach that a single node can have different ones of its visual representations updated based on data received from different software agents.

Regarding claim 38, Dev is silent as to a data conveyance methodology for updating visual representations. Such a conveyance methodology is used to reconcile data conveyed to a display map from different sources or software agents. Dev teaches that a user display is updated via a single server, therefore the problem solved by using the control values detailed in claim 38 is not applicable to Dev.

In light of the above, withdraw of the 35 U.S.C. § 102(b) rejection with regard to claims 1-33 is respectfully requested. Further, Applicants submit that claims 34-38 are also allowable in view of the cited art. Accordingly, Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: 13 Feb 2004

Gregory A. Nelson, Registration No. 30,577
Kevin T. Cuenot, Registration No. 46,283
Brian K. Buchheit, Registration No. 52,667
AKERMAN SENTERFITT
Post Office Box 3188
West Palm Beach, FL 33402-3188
Telephone: (561) 653-5000